WE CLAIM:

1. A method of routing packets intended to be transmitted across a network link protected by a protection path defined by a closed loop of nodes and links through the network, the method comprising the steps of:

determining whether the protected link has failed; and

if the protected link has not failed, sending the packets across the protected link; otherwise, encapsulating the packets within tunnel packets and sending the tunnel packets along the protection path.

2. A method as claimed in claim 1, wherein each packet comprises a header specifying the identity of a source node and a destination node associated with the packet.

3. A method as claimed in claim 2, wherein the source and destination nodes associated with each tunnel packet correspond to the nodes at either end of the protected link.

4. A method as claimed in claim 2, wherein the header of each packet further specifies the nature of the packet as a tunnel packet or a non-tunnel packet.

5. A method as claimed in claim 4, wherein the header of each tunnel packet specifies the identity of the protection path along which it is sent.

6. A method as claimed in claim 5, wherein each tunnel packet further comprises a body and wherein the packet encapsulated by a tunnel packet is contained in the body of the tunnel packet.

7. A method as claimed in claim 6, wherein the encapsulated packet is itself a tunnel packet.

8. A method as claimed in claim 1, wherein the step of determining whether the protected link has failed is performed at a physical electrical layer.

9. A method as claimed in claim 1, wherein the step of determining whether the protected link has failed is performed at a physical optical layer.

10. A method as claimed in claim 1, wherein the step of determining whether the protected link has failed is performed at a logical layer.

11. A method as claimed in claim 10, wherein the logical layer is a SONET STS path.

12. A method as claimed in claim 10, wherein the logical layer is an ATM VCC or VPC.

13. A method as claimed in claim 1, wherein all packets are Internet protocol (IP) datagrams.

14. A method as claimed in claim 1, wherein the trajectory of the protection path is updated dynamically.

15. A method of routing packets received along a network link by a node, each said received packet being associated with a source node and a destination node, the method comprising the steps of:

determining the destination node associated with each received packet;

determining whether the received packet is a tunnel packet encapsulating another packet within its body; and

if the destination node associated with the received packet is the current node and if the received packet is not a tunnel packet, processing the received packet without further forwarding;

if the destination node associated with the received packet is not the current node and if the received packet is not a tunnel packet, forwarding the received packet based on the destination node associated with the received packet;

if the destination node associated with the received packet is the current node and if the received packet is a tunnel packet, retrieving the encapsulated packet from the received packet and forwarding it based on the destination node associated with the encapsulated packet;

if the destination node associated with the received packet is not the current node and if the received packet is a tunnel packet, determining the identity of a protection path along which the tunnel packet was received and forwarding the received packet along a next link in that protection path.

16. A method as claimed in claim 15, wherein any step which involves forwarding a packet across a link protected by a protection path comprises:

determining whether the protected link has failed; and

if the protected link has not failed, sending the packet across the protected link; otherwise, encapsulating the packet within a tunnel packet and sending the tunnel packet along the protection path.

17. A method as claimed in claim 16, wherein each packet comprises a header specifying the identity of a source node and a destination node associated with the packet.

18. A method as claimed in claim 17, wherein the source and destination nodes associated with each tunnel packet correspond to the nodes at either end of the protected link.

19. A method as claimed in claim 17, wherein the header of each packet further specifies the nature of the packet as a tunnel packet or a non-tunnel packet.

20. A method as claimed in claim 19, wherein the header of each tunnel packet specifies the identity of the protection path along which it is sent.

21. A method as claimed in claim 16, wherein all packets are Internet protocol (IP) datagrams.

22. A method of switching traffic in a packet-switched network having a plurality of nodes interconnected by links, the method comprising the steps of:

upon detection of a failure of a link connecting a pair of adjacent nodes, encapsulating packets within the bodies of tunnel packets and forwarding the tunnel packets along a pre-defined protection path which bypasses the failed link.

23. A method as claimed in claim 22, wherein the step of forwarding comprises:

upon receipt of a tunnel packet by one of the adjacent nodes along a protection path, the recipient node retrieving the encapsulated packet and routing it as a function of a destination specified in the header of the encapsulated packet.

24. A method as claimed in claim 23, wherein each packet comprises a header specifying the identity of a source node and a destination node associated with the packet.

25. A method as claimed in claim 24, wherein the source and destination nodes associated with each tunnel packet correspond to the nodes at either end of the protected link.

26. A method as claimed in claim 24, wherein the header of each packet further specifies the nature of the packet as a tunnel packet or a non-tunnel packet.

27. A method as claimed in claim 26, wherein the header of each tunnel packet specifies the identity of the protection path along which it is sent.

28. A method as claimed in claim 23, wherein all packets are Internet protocol (IP) datagrams.

29. A packet-switched network comprising a plurality of nodes interconnected by links, wherein pre-defined protection paths provide protection of a selected plurality of links and wherein adjacent nodes connected by a protected link are adapted to detect a failure of the protected link, to encapsulate packets within tunnel packets, to differentiate between tunnel packets and non-tunnel packets and to exchange the tunnel packets via the protection paths.

30. A packet-switched network as claimed in claim 29, wherein each packet comprises a header specifying the identity of a source node and a destination node associated with the packet.

31. A packet-switched network as claimed in claim 30, wherein the source and destination nodes associated with each tunnel packet correspond to the nodes at either end of the protected link.

32. A packet-switched network as claimed in claim 30, wherein the header of each packet further specifies the nature of the packet as a tunnel packet or a non-tunnel packet.

33. A packet-switched network as claimed in claim 32, wherein the header of each tunnel packet specifies the identity of the protection path along which it is sent.

34. A packet-switched network as claimed in claim 29, wherein all packets are Internet protocol (IP) datagrams.

35. An article of manufacture, comprising:

a computer usable medium having computer readable program code embodied therein for routing packets intended to be transmitted across a network link protected by a protection path defined by a closed loop of nodes and links through the network, the computer readable program code in said article of manufacture comprising:

computer readable program code for causing a computer to determine whether the protected link has failed; and

computer readable program code for causing a computer to send the packets across the protected link if the protected link has not failed; and

computer readable program code for causing a computer to encapsulate the packets within tunnel packets and to send the tunnel packets along the protection path if the protected link has failed.

36. A router for routing packets intended to be transmitted across a network link protected by a protection path defined by a closed loop of nodes and links through the network, comprising:

means for determining whether the protected link has failed; and

means for sending the packets across the protected link if the protected link has not failed; and

means for encapsulating the packets within tunnel packets and for sending the tunnel packets along the protection path if the protected link has failed.

37. An article of manufacture, comprising:

a computer usable medium having computer readable program code embodied therein for routing packets received along a network link by a node, each said received packet being associated with a source node and a destination node, the computer readable program code in said article of manufacture comprising:

computer readable program code for causing a computer to determine the destination node associated with each received packet;

computer readable program code for causing a computer to determine whether the received packet is a tunnel packet encapsulating another packet within its body; and

computer readable program code for causing a computer to process the received packet without further forwarding, if the destination node associated with the received packet is the current node and if the received packet is not a tunnel packet;

computer readable program code for causing a computer to forward the received packet based on the destination node associated with the received packet, if the destination node

associated with the received packet is not the current node and if the received packet is not a tunnel packet;

computer readable program code for causing a computer to retrieve an encapsulated packet from the received packet and forward it based on the destination node associated with the encapsulated packet, if the destination node associated with the received packet is the current node and if the received packet is a tunnel packet; and

computer readable program code for causing a computer to determine the identity of a protection path along which the received packet was received and forward the received packet along a next link in that protection path, if the destination node associated with the received packet is not the current node and if the received packet is a tunnel packet.

38. A router for routing packets received along a network link by a node, each said received packet being associated with a source node and a destination node, comprising:

means for determining the destination node associated with each received packet;

means for determining whether the received packet is a tunnel packet encapsulating another packet within its body; and

means for processing the received packet without further forwarding, if the destination node associated with the received packet is the current node and if the received packet is not a tunnel packet;

means for forwarding the received packet based on the destination node associated with the received packet, if the destination node associated with the received packet is not the current node and if the received packet is not a tunnel packet;

means for retrieving an encapsulated packet from the received packet and forwarding it based on the destination

node associated with the encapsulated packet, if the destination node associated with the received packet is the current node and if the received packet is a tunnel packet; and

means for determining the identity of a protection path along which the received packet was received and forwarding the received packet along a next link in that protection path, if the destination node associated with the received packet is not the current node and if the received packet is a tunnel packet.

39. A protection cycle manager that processes data packets in the event of a failure of a link connected to a routing node, the protection cycle manager comprising:

a packet identifier that identifies, as protection cycle packets, data packets having a specific protection cycle format that includes a packet source and a packet destination and an indication that the packet is a protection cycle packet; and

a packet processor that processes each protection cycle packet to determine whether the packet destination corresponds to the routing node, and:

iii. if the packet destination corresponds to the routing node, the protection cycle packet is treated by the routing node as a data packet received from the packet source via the failed link; and

iv. if the packet destination does not correspond to the routing node, the protection cycle packet is sent to a protection cycle node associated with the routing node.

40. A protection cycle manager as claimed in claim 39, further comprising:

a packeter that converts, in response to failure of a link, affected data packets routed over the failed link into protection cycle packets in the specific protection format.

41. A protection cycle manager as claimed in claim 39, wherein the protection cycle manager further advertises a link failure to the network using a routing protocol.

42. A protection cycle manager as claimed in claim 39, wherein the specific protection cycle format includes a label stack based on Multi-Protocol Label Switching (MPLS).

43. A protection cycle manager as claimed in claim 42, wherein the label stack includes labels for the packet source and the packet destination.

44. A protection cycle manager as claimed in claim 39, wherein the specific protection cycle format includes an IP-in-IP tunnel.

45. A protection cycle manager as claimed in claim 39, wherein the IP-in-IP tunnel includes a header containing the packet source and the packet destination and an indication that the packet is a protection cycle packet.

46. A data router for routing packets intended to be transmitted across a network link protected by a protection path defined by a closed loop of nodes and links through the network, comprising:

a data interface for packets to enter and exit the router; and

a protection cycle packet manager connected to the data interface, for:

i. determining whether the protected link has failed;

ii. sending the packets across the protected link if the protected link has not failed; and

iii. encapsulating the packets within tunnel packets and sending the tunnel packets along the protection path if the protected link has failed.

47. A data router as claimed in claim 46, further comprising:

a packeter that converts, in response to failure of a link, affected data packets routed over the failed link into protection cycle packets in the specific protection format.

48. A data router as claimed in claim 46, wherein the protection cycle manager further advertises a link failure to the network using a routing protocol.

49. A data router as claimed in claim 46, wherein the specific protection cycle format includes a label stack based on Multi-Protocol Label Switching (MPLS).

50. A data router as claimed in claim 49, wherein the label stack includes labels for the packet source and the packet destination.

51. A data router as claimed in claim 46, wherein the specific protection cycle format includes an IP-in-IP tunnel.

52. A data router as claimed in claim 46, wherein the IP-in-IP tunnel includes a header containing the packet source and the packet destination and an indication that the packet is a protection cycle packet.

53. A data router as claimed in claim 46, the protection cycle packet manager being adapted to determine the

destination node associated with each received packet, to determine whether the received packet is a tunnel packet encapsulating another packet within its body and to

i. process the received packet without further forwarding, if the destination node associated with the received packet is the current node and if the received packet is not a tunnel packet;

ii. forward the received packet based on the destination node associated with the received packet, if the destination node associated with the received packet is not the current node and if the received packet is not a tunnel packet;

iii. retrieve an encapsulated packet from the received packet and forward it based on the destination node associated with the encapsulated packet, if the destination node associated with the received packet is the current node and if the received packet is a tunnel packet; and

iv. determine the identity of a protection path along which the received packet was received and forward the received packet along a next link in that protection path, if the destination node associated with the received packet is not the current node and if the received packet is a tunnel packet.